

```

Additional_file_2.txt
#!/usr/bin/perl -w
use Bio::SeqIO;

#####
# Additional_file_2.pl
# author: Leandro Costa do Nascimento
# E-mails: leandro@lge.ibi.unicamp.br or l.costa.nascimento@gmail.com

# Article: A web-based bioinformatics interface applied to Genosoja Project:
# databases and pipelines
# Nascimento et al., 2011
# Bioinformatics - Genomics and Expression Laboratory (LGE)
# http://www.lge.ibi.unicamp.br
# GENOSOJA database: http://www.lge.ibi.unicamp.br/soja

# Usage: perl Additional_file_2.pl <tags_file> <number_bases> <fasta_genome>
# <soap_command>
# Bugs: Probably many! =D
#####

##### Global variables - Don't edit #####
my $genome_file = "";
my $tags_file = "";
my $number_bases = 0;
my $soap_align_command = "";

my %alinhados = ();
my %ok = ();
my %cromossomo_13 = ();
my %alinhamentos_cromossomo_13 = ();
#####

##### Parameters section #####
#####
sub show_parameters{
    print "Usage: perl Additional_file_2.pl <tags_file> <number_bases>
<fasta_genome> <soap_command>\n\n";
    print "tags_file: file with the possible microRNAs in fastq format\n";
    print "number_bases: the script will get bases before and after the alignment
according to this parameter\n";
    exit(0);
}

($tags_file, $number_bases, $genome_file, $soap_align_command) = @ARGV;

if(!defined($tags_file)){
    show_parameters();
}

if(!defined($number_bases)){
    show_parameters();
}

if(!defined($genome_file)){
    show_parameters();
}

if(!defined($soap_align_command)){
    show_parameters();
}
#####
#####
#####

```

```

                                Additional_file_2.txt
### Edit this variables - if you want
#####
my $soap_file = "$tags_file\_X_genome.soap";
my $new = "$tags_file\_X_genome.fasta";
#####
#####

### Soap section
#####
print "Running the soap software to align the reads with the reference\n";
system("$soap_align_command -a $tags_file -D $genome_file.index -o $soap_file -r 2 -v 0");
#####

### Searching for alignments
#####
print "Searching for alignments in the soap output file\n";
open FILE, "<$soap_file";
while(<FILE>){
    chomp;
    my @linha = split(/\t/, $_);

    my $tag = $linha[0];
    my $sinal = $linha[6];
    my $referencia = $linha[7];
    my $position = $linha[8];

    if($referencia eq "Gm13"){
        if(defined($cromossomo_13{$tag})){
            $cromossomo_13{$tag}++;
        }
        else{
            $cromossomo_13{$tag} = 1;
        }

        if($cromossomo_13{$tag} >= 3){
            $alinhamentos_cromossomo_13{$tag} = "";
            next;
        }
        else{
            my $temp = $cromossomo_13{$tag};

            if(defined($ok{$tag})){
                $temp += $ok{$tag};
            }

            if(defined($alinhamentos_cromossomo_13{$tag})){
                $alinhamentos_cromossomo_13{$tag} .=
                ";$tag\_$temp,$position,$sinal";
            }
            else{
                $alinhamentos_cromossomo_13{$tag} =
                "$tag\_$temp,$position,$sinal";
            }
        }
    }
    else{
        if(defined($ok{$tag})){
            $ok{$tag]++;
        }
        else{
            $ok{$tag} = 1;
        }

        if(defined($alinhados{$referencia})){
            $alinhados{$referencia} .= ";$tag\_$ok{$tag},$position,$sinal";
        }
    }
}

```

```

                                Additional_file_2.txt
}
else{
    $alinhados{$referencia} = "$tag\$_ok{$tag},$position,$sinal";
}
}
}
close FILE;

foreach(keys %alinhamentos_cromossomo_13){

if($alinhamentos_cromossomo_13{$_} ne ""){
    if(defined($alinhados{Gm13})){
        $alinhados{Gm13} .= ";$alinhamentos_cromossomo_13{$_}";
    }
    else{
        $alinhados{Gm13} = "$alinhamentos_cromossomo_13{$_}";
    }
}
#####
#####
#####

### Getting the final sequences
#####
print "Getting the final sequences\n";
my $inseq = Bio::SeqIO-> new(-file => "<$genome_file", -format => "fasta" );
while (my $seq = $inseq->next_seq){
    my $agora = $seq->display_id;

    if(defined($alinhados{$agora})) {
        my @split = split(/\;/, $alinhados{$agora});

        # running in the separation of the tags by ";"
        foreach(@split){
            # running in the separation of the tag name and position by ","
            my @array = split(/\,/, $_);

            my $inicio = $array[1] - $number_bases;
            my $fim = $array[1] + $number_bases;
            my $tamanho = $seq->length;

            if($inicio < 1){
                $inicio = 1;
            }

            if($fim > $tamanho){
                $fim = $tamanho;
            }

            my $new_seq = $seq->subseq($inicio, $fim);
            open NEW, ">$new";
                # tag reference:initial position in the reference.end position in
                # the reference alignment direction
                print NEW ">$array[0] $agora:$array[1] $array[2]\n";
                print NEW "$new_seq\n";
            close NEW;
        }
    }
}
#####
#####
#####

```